Time-Aware Probabilistic Knowledge Graphs

Melisachew Wudage Chekol
Data and Web Science Group, University of Mannheim, Germany
mel@informatik.uni-mannheim.de

Heiner Stuckenschmidt
Data and Web Science Group, University of Mannheim, Germany
heiner@informatik.uni-mannheim.de

Abstract

The emergence of open information extraction as a tool for constructing and expanding knowledge graphs has aided the growth of temporal data, for instance, YAGO, NELL and Wikidata. While YAGO and Wikidata maintain the valid time of facts, NELL records the time point at which a fact is retrieved from some Web corpora. Collectively, these knowledge graphs (KG) store facts extracted from Wikipedia and other sources. Due to the imprecise nature of the extraction tools that are used to build and expand KG, such as NELL, the facts in the KG are weighted (a confidence value representing the correctness of a fact). Additionally, NELL can be considered as a transaction time KG because every fact is associated with extraction date. On the other hand, YAGO and Wikidata use the valid time model because they maintain facts together with their validity time (temporal scope). In this paper, we propose a bitemporal model (that combines transaction and valid time models) for maintaining and querying bitemporal probabilistic knowledge graphs. We study coalescing and scalability of marginal and MAP inference. Moreover, we show that complexity of reasoning tasks in atemporal probabilistic KG carry over to the bitemporal setting. Finally, we report our evaluation results of the proposed model.

2012 ACM Subject Classification Information systems → Web Ontology Language (OWL); Computing methodologies → Probabilistic reasoning; Computing methodologies → Temporal reasoning

Keywords and phrases temporal, probabilistic, knowledge graph, OWL-RL

Digital Object Identifier 10.4230/LIPIcs.TIME.2019.8

1 Introduction

Temporal databases have been studied extensively [44, 15, 31]. Recently, support for temporal data from database vendors, such as Teradata, Oracle DB, IBM DB2, PostgreSQL, and so on, has been growing. On the other hand, probabilistic temporal databases have been given little attention [32, 12, 11, 8] and even less for probabilistic temporal knowledge graphs [6, 13]. Several Web knowledge graphs, such as YAGO [23], Wikidata [43], NELL [3], and DBpedia [1], already contain temporal data (each fact is associated with a valid time). In particular, NELL contains temporal probabilistic data where each fact is associated with a transaction time (the time when a fact is extracted and stored in a knowledge graph). Besides, the emergence of open information extraction as a tool for constructing and expanding knowledge graphs (KG) has aided the growth of temporal data [36, 35, 41]. In addition to valid time support, maintaining the transaction time of facts is relevant because never-ending open information extraction systems such MinIE [17], NELL, Google's Knowledge Vault [10], Microsoft's Satori continuously learn new facts from the Web and it is important to record the date on which a fact is learned. This is evident from the NELL knowledge graph as they keep track of learned dates in terms of iterations. Hence, transaction times represent the date on which a fact is extracted or recorded. The valid time indicates the time period on which a fact is considered valid or true. Furthermore, when such facts are learned/extracted from open text,

26th International Symposium on Temporal Representation and Reasoning (TIME 2019).
Editors: Johann Gamper, Sophie Pinchinat, and Guido Sciavicco; Article No. 8; pp. 8:1–8:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

they are associated with some confidence score. We need a model that supports these three desirable aspects (transaction time, validity time, and confidence scores). Thus, in this work, we propose a framework for representing and querying bitemporal probabilistic KG.

Most existing KG contain schema that can be represented by lightweight ontology languages such as the OWL profiles (for instance, NELL's schema can be captured by the web ontology rule language called OWL RL [26]). In this paper, we study a bitemporal probabilistic extension of OWL RL as modeling language for KG.

Probabilistic graphical models have been widely used to reason about facts extracted at Web scale using a combination of hand-crafted and extracted inference rules [37]. In particular, Markov logic networks (MLN) can be used to deal with temporal relations in open information extraction [28] or checking the consistency of knowledge bases [6, 7]. MLN extends first order logic with uncertainty by attaching weights to formulas. In MLN, there are two important reasoning tasks, marginal and maximum a-posteriori (MAP) inference, in MLN. The former computes the probability of a set of random variables (temporal facts in our setting) whereas the later computes the most probable and consistent world (temporal knowledge graph). Since marginal inference does not scale well, in this work, we present a novel approximate algorithm to compute the marginal distributions of temporal queries efficiently.

In bitemporal probabilistic KG, it is necessary to remove redundant facts, known as deduplication, so as to avoid errors in query answers and reduce the size of the graph. Hence, we embark upon a challenging problem in coalescing under uncertainty. It amounts to merging facts with identical non-temporal arguments and adjacent or overlapping time-intervals. For instance, consider the deterministic facts $\langle r(a,b,2,5), 0.8\rangle$ and $\langle s(a,b,4,7), 0.7\rangle$ as well as the axiom $r \sqsubseteq s$, clearly the two facts cannot be coalesced with the well known sort-merge approach [2]. Hence, we need to first perform inference using the axiom so that we utilize sort-merge to get the coalesced fact $s(a,b,2,7)$. However, we still need to determine the weight of the coalesced fact. For this task, we propose a number of approaches including a rule-based algorithm for coalescing that uses marginal inference to determine the weight of the coalesced fact.

Overall, the contributions of this paper are: (i) we propose a bitemporal model for Web knowledge graphs by considering OWL RL as an atemporal ontology language, (ii) we extend it (bitemporal KG) using MLN for modeling uncertainty, (iii) we address temporal coalescing (both in data and queries) in a probabilistic setting, (iv) we present a novel N-hop based approximate algorithm for marginal inference, (v) we show that the bitemporal scoping of probabilistic facts does not introduce any complexity overhead, and (vi) we provide an empirical evaluation of the proposed approach over the Wikidata KG.

1.1 Motivating Example

To motivate the purpose of this study we rely on two prominent knowledge graphs: NELL and Wikidata. NELL records the extraction dates of facts as shown in Table 1. For instance, it is extracted or recorded on 09/02/2017 that Fernando Torres plays for the Chelsea football club with a confidence of 96.9%. This shows that NELL's representation model resembles that of transaction time from relational databases. On the other hand, Wikidata scopes temporally some of the facts, for instance, with 100% confidence Fernando Torres played for Atletico Madrid from 2001 to 2007 before rejoining them on 2016 and he is still playing for them. This shows that Wikidata uses the valid time model for modeling temporal information. However, Wikidata does not record the extraction date of facts, conversely, NELL does not maintain the valid time of facts. Thus, what is missing is a model which combines both

Table 1 NELL and Wikidata representations of the career of Fernando Torres.

NELL: learned fact	Date learned	Confidence
(*torres*, *type*, *athlete*)	12/01/2010	100.0
(*torres*, *stadium*, *anfield*)	10/11/2015	100.0
(*torres*, *playsfor*, *chelsea*)	09/02/2017	96.9
(*torres*, *playsfor*, *spain*)	08/08/2011	87.5
Wikidata: extracted fact	Valid time	Confidence
(*torres*, *playsfor*, *atleticoMadrid*)	$[2001, 2007)$	100.0
(*torres*, *playsfor*, *liverpool*)	$[2007, 2011)$	100.0
(*torres*, *playsfor*, *chelsea*)	$[2011, 2015)$	100.0
(*torres*, *playsfor*, *atleticoMadrid*)	$[2016, now)$	100.0

transaction time and valid time (i.e., bitemporal). For instance, the fact that "Fernando Torres played for Chelsea football club from 2011 to 2015" is extracted on 09/02/2017, can be modeled as a bitemporal fact as shown below:
(*torres*, *playsfor*, *chelsea*, [2011,2015), [09/02/2017,*UC*), 96.9), where *UC* is short for until changed – when the end date of the transaction or recording date is unknown.

In addition to temporal probabilistic facts, a KG can contain (temporal) inference rules, for instance, some of the deduction rules learned from the NELL KG using ProbFOIL$^+$ [33] are shown below: (i) if an athlete led a sports team, then she probably plays for that team; (ii) if an athlete led a sports team and her team plays against another team, then she probably plays for that team; and (iii) if an athlete plays some sport and a team plays that sport, then that athlete probably plays for that team. These rules can be converted into temporal inference rules by adding temporal variables to the predicates and using a numerical predicate that tests interval overlap.

	Inference rule	Weight
i.	$athleteledsportsteam(a, b) \rightarrow playsfor(a, b)$	0.93
ii.	$athleteledsportsteam(a, y), teamplaysagainstteam(b, y) \rightarrow playsfor(a, b)$	0.96
iii.	$athleteplayssport(a, y), teamplayssport(b, y) \rightarrow playsfor(a, b)$	0.93

In order to perform reasoning tasks over probabilistic temporal facts and temporalized inference rules (see Section 4), in this study, we propose a bitemporal model for probabilistic knowledge graphs.

2 Background

2.1 OWL RL

A knowledge graph (KG) is a set of triples that can be be encoded in the W3C standard RDF data model [22]. Let I and L be two disjoint sets denoting the set of IRIs (identifying resources) and literals (character strings or some other type of data), respectively. We abbreviate the union of these sets $(\mathsf{I} \cup \mathsf{L})$ as IL. A triple has the form $(s, r, o) \in \mathsf{I} \times \mathsf{I} \times \mathsf{IL}$ where s is the *subject*, r is the *predicate* or relation, and o is the *object* of the triple. A KG can be extended with temporal information by labeling each triple in the graph with a temporal element. For instance, the temporal element can represent the time period in which the triple is valid, i.e., the *valid time* of the triple [29, 18, 20]. KG often contain an ontology or schema that is encoded in some lightweight language such as OWL RL.

OWL RL is a tractable rule-based ontology language. It prohibits the use of disjunctions of classes and existential quantification for superclass expression to enable polynomial time reasoning [26]. We use description logic for concisely expressing the syntax of OWL RL axioms. We denote a set of concept names by $\mathsf{N_C} \subseteq \mathsf{I}$; roles names by $\mathsf{N_R} \subseteq \mathsf{I}$; individual names by $\mathsf{N_I} \subseteq \mathsf{IL}$; and their union by N i.e., $\mathsf{N} \subseteq \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}$. We do not consider datatypes for the sake of space, however, our work can be easily extended (for instance by following the work in [5]). Additionally, we use the following notations through out the paper: a and b denote *instance or individual names* in $\mathsf{N_I}$; A and B denote *class or concept names* in $\mathsf{N_C}$; C (resp. D) is a *complex class expression* denoting *subclass expression* (resp. *superclass expression*); p, p^-, r, r_1, r_2 are *role names* in $\mathsf{N_R}$; s and s_i denote axioms or assertions. An OWL RL knowledge graph contains a set of axioms. The syntax of which is given by the following grammar:

$$
\begin{aligned}
Axiom, \mathsf{s} &::= C \sqsubseteq D \mid r_1 \sqsubseteq r_2 \mid r_1 \circ r_2 \sqsubseteq r \mid A(a) \mid r(a,b) \\
C &::= A \mid \bot \mid \{a\} \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \\
D &::= A \mid \bot \mid \neg D \mid D \sqcap D \mid \forall r.D \mid \exists r.\{a\} \mid \leqslant 1\ r.C \mid \leqslant 0\ r.C \\
r, r_1, r_2 &::= p \mid p^-
\end{aligned}
$$

We refer to axioms of the form $A(a)$ and $r(a,b)$ *instance assertions* (facts) and those that appear under the subsumption ($\sqsubseteq$) relation (for instance $r_1 \sqsubseteq r_2$) are called *inclusion axioms*. The assertion $A(a)$ (resp. $r(a,b)$) can be written in RDF syntax as $(a, \texttt{type}, A)$ (resp. (a, r, b)). In this paper, we consider a temporal extension of OWL RL where the instance assertions are temporal and the inclusion axioms are atemporal. We refer to temporal instance assertions as *temporal facts*. In order to facilitate the transformation of OWL RL axioms into first-order formulas, we consider the following normal forms [26]:

$$
\begin{array}{llllll}
C(a) & r(a,b) & A \sqsubseteq C & A \sqcap B \sqsubseteq C & r_1 \sqsubseteq r_2 & \\
A \sqsubseteq \{a\} & A \sqsubseteq \leqslant 1\ r.C & A \sqsubseteq \forall r.C & \{a\} \sqsubseteq C & r_1 \circ r_2 \sqsubseteq r \quad r_1^- \sqsubseteq r_2 & dis(r_1, r_2)
\end{array}
$$

The axiom $dis(r_1, r_2)$ denotes that r_1 and r_2 are disjoint. Note that the axiom $r_1^- \sqsubseteq r_2$ is subsumed by the axiom $r_1 \sqsubseteq r_2$. An OWL RL KG is in *normal* form if its axioms are normalized. Throughout this paper, we assume that the axioms of KG are normalized. The normal forms are obtained by leveraging structural transformation. Although, the OWL specification provides a partial axiomatization of the OWL RDF-based semantics using a fixed set of OWL RL/RDF rules[1]. In this study, we focus on the OWL direct semantics and translate the OWL RL inference rules into first-order formulas. However, our approach is also applicable to OWL RDF-based semantics. A probabilistic extension of an OWL RL KG can be efficiently modeled using Markov logic network. Although, in this paper, we use Markov logic network, our approach allows to easily adapt other probabilistic modeling frameworks such as ProbLog and probabilistic soft logic.

2.2 Markov Logic Network

A Markov logic network (MLN) combines Markov networks and first-order logic (FOL) by attaching weights to first-order formulas. An MLN program $\mathcal{L}$ is a set of pairs $\mathcal{L} = (f_i, w_i)$ where f_i is a FOL formula and w_i is a real number representing its weight [34]. In this paper, we use the Horn fragment of FOL which efficiently represents OWL RL inference rules. For brevity, we will drop $\forall$ quantifier from all the formulas. The probabilistic facts and rules given in the motivating example can be seen as an MLN program.

[1] `https://www.w3.org/TR/owl2-profiles/`

Together with a set of constants $\mathcal{C}$, an MLN defines a Markov network $M_{\mathcal{L},\mathcal{C}}$, where $M_{\mathcal{L},\mathcal{C}}$ contains one node for each possible grounding of each predicate appearing in $\mathcal{L}$. The value of the node is 1 if the ground predicate is true, and 0 otherwise. The probability distribution over a possible world x, specified by a ground Markov network $M_{\mathcal{L},\mathcal{C}}$, is:

$$P(X = x) = Z^{-1}\exp\big(\sum_i w_i n(f_i, x)\big),$$

where $n(f_i, x)$ is the number of true groundings of f_i in x. The groundings of a formula are formed simply by replacing its variables with constants in all possible ways. A ground MLN can be turned into a factor graph. A *factor graph* is a set of factors $\Phi = \{\phi_1, \ldots, \phi_n\}$ where each factor ϕ is a function $\phi(X)$ over a set of random variables X. A factorization of a function g over the variables X is given by: $g(X) = \prod_{i=1}^{n} \phi_i(X_i)$. We convert a ground MLN into a factor graph by using the following: each ground atom $p_i(a)$ in an MLN becomes a random variable X_i, and each ground formula (f_i, w_i) becomes a factor $\phi(X_i)$ which has a value e^{w_i} if the formula is true and 1 otherwise. Such a factor graph determines a probability distribution over X,

$$P(X = x) = Z^{-1}\prod_i \phi(X_i) = Z^{-1}\exp\big(\sum_i w_i n_i(f_i, x)\big).$$

There are two important reasoning tasks in MLN. The first one is called *marginal* inference which is the the task of computing the probability of a set of variables given evidence. The complexity of this problem is known to be #P-hard. The second one is *maximum a-posteriori* (MAP) inference which is the task of finding *the most probable state* of the world, i.e., finding a complete assignment to all ground atoms which maximizes the probability. This problem is known to be NP-hard. We will study these inference tasks for bitemporal probabilistic KG but first we present bitemporal KG.

3 Bitemporal Knowledge Graphs

A bitemporal KG is an extension of a conventional KG by adding temporal elements to each instance assertion in the graph (similar to bitemporal databases [27]). A fact in a bitemporal KG is timestamped with time intervals that represent the fact's *valid time* and *transaction time*. A valid time is the time period in which a fact is considered true or valid. Transaction time is the time when a fact is added to a KG. Thus, a bitemporal KG needs domains for two temporal universes, the valid time universe and the transaction time universe, and it may be desirable or convenient to restrict them to some subset of T. Therefore, let $\mathsf{T_v} \subseteq \mathsf{T}$ denote the valid time universe of a bitemporal KG, and $\mathsf{T_t} \subseteq \mathsf{T}$ denote its transaction time universe. We consider $\mathsf{T} = \mathsf{T_v} \cup \mathsf{T_t}$ to be a discrete time domain which is a linearly ordered finite sequence of *time points*; for instance, days, minutes, or milliseconds. The finite domain assumption ensures that there are finitely many possible worlds in the probabilistic extension. A *time interval* is an ordered pair $[t_b, t_e)$ of time points, with $t_b \leq t_e$ and $t_b, t_e \in \mathsf{T}$, which denotes the closed-open interval of time points from t_b to t_e[2]. We will work with the interval-based temporal domain to define our data model.

[2] It is possible to extend to other interval based representations such as $[t_b, t_e]$, left and right closed interval.

▶ **Definition 1** (Bitemporal KG). *A bitemporal KG is a tuple $G = \langle \mathcal{S}, \mathcal{A} \rangle$ where $\mathcal{S}$ is the atemporal component representing the schema part (in OWL RL) and $\mathcal{A}$ is a set of OWL RL instance assertions in which each assertion $A(a)$ (resp. $r(a,b)$) in the graph is associated with a valid time $[v_b, v_e) \in \mathsf{T_v}$ and transaction time $[t_b, t_e) \in \mathsf{T_t}$, i.e., $g = A(a, [v_b, v_e), [t_b, t_e))$ (resp. $g = r(a, b, [v_b, v_e), [t_b, t_e))$). We refer to g as a bitemporal fact and write it as a first order predicate $\textsc{ca}(a, A, v_b, v_e, t_b, t_e)$ or $\textsc{ra}(a, r, b, v_b, v_e, t_b, t_e)$ with* CA *for concept assertion and* RA *for role assertion.*

Right-unlimited time intervals are expressed as $[t_b, \mathit{UC})$ for transaction time and $[v_b, \mathit{now})$ for valid time, where *UC* is short for Until Changed and *now* denotes the current time instance. Note that both *UC* and *now* are replaced by the current time during reasoning. For the sake of presentation, we remove the day and month of a given date and write just the years for both valid and transaction time intervals.

▶ **Example 1.** *We convert some of the facts of the KG in Table 1 into bitemporal facts as shown below. This is the same as aligning (or loosely merging) the NELL and Wikidata KG for the task of KG completion or bitemporal KG construction.*

$ra(\mathit{torres}, \mathit{playsfor}, \mathit{chelsea}, 2011, 2015, 2017, \mathit{UC})$
$ra(\mathit{torres}, \mathit{stadium}, \mathit{anfield}, 2007, 2011, 2015, \mathit{UC})$
$ra(\mathit{torres}, \mathit{playsfor}, \mathit{atleticoMadrid}, 2001, 2007, 2016, \mathit{UC})$

Bitemporal KG expansion. Relying on the above example, we can use a rule-based approach for KG expansion. For instance, using the NELL KG, we can extend Wikidata with the help of rules (a rule per relation) of the following form:

$$\mathit{playsfor}(x, y, t_b, t_e), \mathit{playsfor}(x, y, v_b, v_e) \rightarrow \mathit{playsfor}(x, y, v_b, v_e, t_b, t_e).$$

It is possible to use more complex rules that include predicates which test the semantic similarity of relation names. However, this is beyond the scope of this work.

For a bitemporal KG G, its *snapshot* at a valid time v and a transaction time t is the graph $G(v,t)$ (the non-temporal KG): $G(v,t) = \{A(a) \mid A(a, [v,v), [t,t)) \in G\} \cup \{r(a,b) \mid r(a, b, [v,v), [t,t)) \in G\}$. The atemporal or non-temporal KG associated with a bitemporal KG is $u(G) = \bigcup_{v,t} G(v,t)$, the union of the graphs $G(v,t)$. Relying on this characterization, we define temporal entailment.

▶ **Definition 2** (Bitemporal entailment). *We define temporal entailment as follows: for two bitemporal knowledge graphs G_1 and G_2, $G_1 \models_{v,t} G_2$ if and only if $G_1(v,t) \models G_2(v,t)$ for each v and t; $\models_{v,t}$ denotes bitemporal entailment and $\models$ is the standard OWL RL entailment [26].*

Alternatively, bitemporal entailment can be reduced into temporal entailment defined in [20]. For two bitemporal graphs G_1 and G_2, let the valid graphs be $G_1(v) = \{A(a, [t_b, t_e)) | A(a, [v,v), [t_b, t_e)) \in G\}$ and similarly $G_2(v)$; and transaction graphs be $G_1(t) = \{A(a, [v_b, v_e)) | A(a, [v_b, v_e), [t,t)) \in G\}$ and similarly $G_2(t)$. $G_1 \models G_2$ iff $G_1(v) \models_t G_2(v)$ and $G_1(t) \models_v G_2(t)$ where $\models_t$ and $\models_v$ denote temporal entailment.

▶ **Definition 3.** *For a bitemporal KG G, a translation of the normal forms of OWL RL axioms into FOL predicates is given by a bijective function φ as shown.*

$A \sqsubseteq C$	$\mapsto$	$\textsc{sc}(A, C)$	$r_1^- \sqsubseteq r_2$	$\mapsto$	$\textsc{inv}(r_1, r_2)$
$A \sqcap B \sqsubseteq C$	$\mapsto$	$\textsc{int}(A, B, C)$	$dis(r_1, r_2)$	$\mapsto$	$\textsc{dis}(r_1, r_2)$
$A \sqsubseteq \{a\}$	$\mapsto$	$\textsc{sc}(A, a)$	$A(a)$	$\mapsto$	$\textsc{ca}(a, A)$
$A \sqsubseteq \forall r.C$	$\mapsto$	$\textsc{all}(A, r, C)$	$\{a\} \sqsubseteq C$	$\mapsto$	$\textsc{ca}(a, C)$
$A \sqsubseteq \leqslant 1 r.C$	$\mapsto$	$\textsc{atmost}(A, r, C)$	$r(a,b)$	$\mapsto$	$\textsc{ra}(a, r, b)$
$r_1 \sqsubseteq r_2$	$\mapsto$	$\textsc{sp}(r_1, r_2)$	$a \in \mathsf{N_I}$	$\mapsto$	$\textsc{nom}(a)$
$r_1 \circ r_2 \sqsubseteq r$	$\mapsto$	$\textsc{rcomp}(r_1, r_2, r)$			

We propose a rule-based instance retrieval in which we compute all entailments of the form $\textsc{ca}(a, A, v_b, v_e, t_b, t_e)$ and $\textsc{ra}(a, r, b, v_b, v_e, t_b, t_e)$ for a given KG. For this task, we use the temporal OWL RL inference rules shown in Figure 1. These rules are applied repeatedly, to a given KG, until no new conclusion can be made. It is worth noting that the inference rules do not materialize assertions that can be deduced if the KG is inconsistent. Nevertheless, inconsistencies lead to entailments of the form $\textsc{ca}(a, \bot, v_b, v_e, t_b, t_e)$ for some constant a.

▶ **Lemma 1.** *Let G be a bitemporal KG with an OWL RL schema, its closure can be computed, by applying the rules in Figure 1 until closure, in polynomial time.*

The above lemma follows from the results in OWL RL [26]. Note that in this work, we use FOL syntax of KG axioms, assertions and inference rules for brevity. However, it is also possible to use *reification* in order to represent bitemporal KG using the RDF syntax. Alternatively, more compact representations based on RDR (reification done right) can be utilized [21]. In addition, a more detailed discussion, on the syntax of temporal KG, can be found in [4]. As an example, the first temporal fact in Example 1, can be represented in RDR syntax as follows:

```
<<torres playsfor chelsea>> beginValid 2011 ;
                            endValid 2015 ;
                            beginTrans 2017 ;
                            endTrans UC .
```

In the above syntax, beginValid, endValid, beginTrans, and endTrans are vocabularies denoting the start and end points of valid and transaction time intervals. A query language for bitemporal KG can be defined by extending query languages for valid time KG such as SPARQ-LTL [4]. For the sake of space, we do not present a query language for bitemporal KG.

4 Bitemporal Probabilistic Knowledge Graphs

In this section, we propose a data model for bitemporal probabilistic KG.

4.1 Data Model

A bitemporal probabilistic KG contains a set of temporally annotated facts each with a confidence weight. A formal definition is provided below.

▶ **Definition 4.** *A bitemporal probabilistic KG is a tuple $K = (\mathcal{S}, \mathcal{A}, \mathcal{X} \cup \mathcal{R})$ where $\mathcal{S}$ is a set of OWL RL axioms, $\mathcal{A} = \{(g, w_1), \ldots, (g_n, w_n)\}$ is a set of bitemporal facts in which $g_i \in \mathcal{A}$ has a confidence w_i; $\mathcal{R}$ is temporal OWL RL inference rules given in Figure 1, each rule r_i is deterministic and has weight $w_i = \infty$; and $\mathcal{X} = \{(f_1, w_1), \ldots, (f_m, w_m)\}$ is a set of OWL RL axioms representing background knowledge where each w_j denotes the weight of formula f_i; and $\mathcal{F} = \mathcal{X} \cup \mathcal{R}$.*

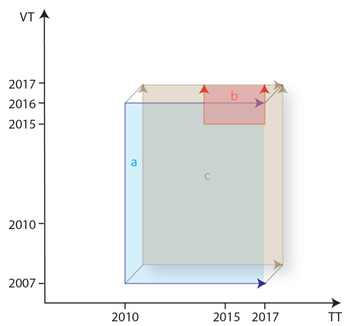
Given a bitemporal probabilistic KG $K = (\mathcal{S}, \mathcal{A}, \mathcal{X} \cup \mathcal{R})$, the probability of a possible world $K' = (\mathcal{S}', \mathcal{A}', \mathcal{X} \cup \mathcal{R}) \subseteq K$ is given by the following log-linear distribution:

$$P(K') = Z^{-1} \exp\Big(\sum_{(f_i, w_i) \in \mathcal{X}' : K \models_{v,t} f_i} w_i n(f_i, K') \Big), \quad Z = \sum_{K_j \subseteq K} \exp\big(\sum_i w_i n(f_i, K_j) \big);$$

where $n(f_i, K')$ is the number of groundings of the formula f_i that evaluate to true in the world K', Z is the normalization constant, and $\models_{v,t}$ denotes bitemporal entailment.

$$
\begin{aligned}
&(r_1)\quad \textsc{sc}(a,b), \textsc{ca}(x,a,v,t) \rightarrow \textsc{ca}(x,b,v,t).\\
&(r_2)\quad \textsc{int}(a,b,c), \textsc{ca}(x,a,v^1,t^1), \textsc{ca}(x,b,v^2,t^2), \textsc{ov}(v^1,v^2,t^1,t^2) \rightarrow \textsc{ca}(x,c,v,t).\\
&(r_3)\quad \textsc{all}(a,r,c), \textsc{ca}(x,a,v^1,t^1), \textsc{ra}(x,r,y,v^2,t^2), \textsc{ov}(v^1,v^2,t^1,t^2) \rightarrow \textsc{ca}(y,c,v,t).\\
&(r_4)\quad \textsc{atmost}(a,r,c), \textsc{ca}(x,a,v^1,t^1), \textsc{ra}(x,r,y,v^2,t^2),\\
&\qquad\quad \textsc{ca}(y,c,v^3,t^3), \textsc{ra}(y,r,z,v^4,t^4), \textsc{ca}(z,c,v^5,t^5), \textsc{ov}(v^1,t^1,\ldots,t^5,t^5) \rightarrow \mathit{equal}(y,z,v,t).\\
&(r_5)\quad \textsc{sp}(r,s), \textsc{ra}(x,r,y,v,t) \rightarrow \textsc{ra}(x,s,y,v,t).\\
&(r_6)\quad \textsc{rcomp}(r,s,t), \textsc{ra}(x,r,y,v^1,t^1), \textsc{ra}(y,s,z,v^2,t^2), \textsc{ov}(v^1,t^1,v^2,t^2) \rightarrow \textsc{ra}(x,t,z,v,t).\\
&(r_7)\quad \textsc{inv}(r,s), \textsc{ra}(y,r,x,v,t) \rightarrow \textsc{ra}(x,s,y,v,t).\\
&(r_8)\quad \textsc{dis}(r,s), \textsc{ra}(a,r,b,v^1,t^1), \textsc{ra}(a,s,b,v^2,t^2), \textsc{ov}(v^1,t^1,v^2,t^2) \rightarrow \textsc{ca}(a,\bot,v^1,t^1).\\
&(r_9)\quad \textsc{ca}(x,a,v,t) \rightarrow \textsc{ca}(x,\top,v,t).\\
&(r_{10})\quad \textsc{ca}(x,a,v,t), \textsc{nom}(a) \rightarrow \textsc{ca}(a,x,v,t).\\
&(r_{11})\quad \textsc{ca}(x,a,v,t), \textsc{nom}(a) \rightarrow \textsc{nom}(x).\\
&(r_{12})\quad \textsc{ra}(x,r,y), \textsc{ca}(z,y,v,t), \textsc{nom}(y) \rightarrow \textsc{ra}(x,r,z,v,t).\\
&(r_{13})\quad \neg\textsc{ca}(x,\bot,v,t).
\end{aligned}
$$

Figure 1 *Notation*: $v^i = [v^i_b, v^i_e), v = [v_b, v_e), t^i = [t^i_b, t^i_e)$, and $t = [t_b, t_e)$. A temporalized variant of OWL RL inference rules that we denote by $\mathcal{R}$. They are partially drived from the materialization calculus in [26]. All of the formulas are universally quantified over all the variables. OV is short for overlaps, it checks if a given set of intervals are overlapping. $\top$ and $\bot$ are constant symbols representing top and bottom concepts. Note that rule r_{13} does not belong to the inference rules for OWL RL. This rule takes the notion of inconsistency into account.



a $\langle \mathit{livesIn}(\mathit{John}, \mathit{Paris}, [2007, 2016), [2010, \mathit{now})), 0.9\rangle$
b $\langle \mathit{livesIn}(\mathit{John}, \mathit{Paris}, [2015, \mathit{now}), [2015, \mathit{now})), 0.5\rangle$
c $\langle \mathit{livesIn}(\mathit{John}, \mathit{Paris}, [2007, \mathit{now}), [2010, \mathit{now})), ?\rangle$

Figure 2 A graphical representation of coalescing bitemporal probabilistic facts. The X-axis represents transaction time (TT) and the Y-axis is valid time (VT). The goal is to determine the weight "?" of the coalesced fact **c**.

An important problem in temporal databases is coalescing, we investigate this problem in a probabilistic setting for bitemporal KG.

4.2 Coalescing and Deduplication

Coalescing is a technique used in temporal databases for duplicate elimination [9]. Coalescing is the process of merging bitemporal facts with identical non-temporal arguments and adjacent or overlapping time-intervals. Hence, it is important for deduplication. In other words, coalescing is useful for: reducing the size of probabilistic temporal KG, and preventing incorrect answers in query evaluation. For instance, consider the query "*does John live in*

Paris from 2014 to 2017?" on the temporal facts of Figure 2 before coalescing, i.e., (a) and (b). The result is no, however, the same query on the coalesced fact (c) (brown part of the figure), returns yes. Uncoalesced facts can arise in various cases: during query evaluation via projection or union operations, by not enforcing coalescing in update or insertion operations, and through information extraction from diverse sources or accuracy of the extractor.

A bitemporal KG K is called duplicate-free, if for all pairs of facts $\text{RA}(a, r, b, v_b, v_e, t_b, t_e)$, $\text{RA}(a, r, b, v'_b, v'_e, t'_b, t'_e) \in K$, it holds that: $[v_b, v_e) \cap [v'_b, v'_e) = \emptyset$ and $[t_b, t_e) \cap [t'_b, t'_e) = \emptyset$. In other words, if the non-temporal terms of two temporal facts are the same, then their temporal terms must be disjoint (non-overlapping). Coalescing is challenging in a probabilistic setting, because it is not clear what should be the weight of the new (coalesced) fact. In order to coalesce bitemporal facts, either the valid time or transaction time intervals must be overlapping. The weight of the coalesced fact can be computed using a number of approaches that are shown below. For the sake of space, we will not present a comparison of the approaches. Instead, for our purpose, we use a rule-based approach that uses probabilistic inference in order to compute the exact weight of a coalesced fact.

1. Max: $w_3 = max(w_1, w_2)$, if two facts can be coalesced, assign the higher weight of the two to the coalesced fact,
2. Average: $w_3 = (w_1 + w_2)/2$,
3. Weighted-Average: $w_3 = (\ell_1 w_1 + \ell_2 w_2)/(\ell_1 + \ell_2)$, where ℓ_1 and ℓ_2 are the lengths of the intervals of fact 1 and fact 2,
4. Min: $w_3 = min(w_1, w_2)$ this is similar to Godel's fuzzy conjunction operator,
5. Lukasiewicz's relaxation: this approach works if the weights are between 0 and 1, $w_3 = max(0, w_1 + w_2 - 1)$, and
6. Rule-based coalescing with marginal inference: in order to coalesce the facts of a bitemporal probabilistic KG K, we can construct Horn rules for each relation in the KG, i.e., for each concept or relation m_i in K, we create a rule of the following form:

$$
\begin{aligned}
(r_{14})\quad & \text{CA/RA}(x, m_i, y, v_b, v_e, t_b, t_e), \text{CA/RA}(x, m_i, y, v'_b, v'_e, t'_b, t'_e),\\
& v = min(v_b, v'_b), v' = max(v_e, v'_e), v_b \leq v'_e,\ v'_b \leq v_e,\\
& t = min(t_b, t'_b), t' = max(t_e, t'_e), t_b \leq t'_e,\ t'_b \leq t_e \rightarrow \text{CA/RA}(x, m_i, y, v, v', t, t').
\end{aligned}
$$

The expression $v_b \leq v'_e,\ v'_b \leq v_e$ (resp. $t_b \leq t'_e,\ t'_b \leq t_e$) tests temporal overlap of the intervals $[v_b, v_e)$ and $[v'_b, v'_e)$ (resp. $[t_b, t_e]$ and $[t'_b, t'_e]$). Besides, *min*, and *max* are predicates representing minimum, and maximum functions respectively. The weights of the coalesced facts are determined by performing marginal inference on the given KG. Once these weights are determined, the uncoalesced facts are removed from the KG. The approach uses one rule for each relation. If a KG has several thousands of relations, we need the same number of coalescing rules to remove duplicates from the KG. Hence, this operation can be very expensive, however, it is done only once. Our rule-based coalescing procedure is given in Algorithm 1.

▶ **Example 2.** *Consider coalescing the bitemporal probabilistic facts, (a) and (b), shown in Figure 2. This operation merges the two facts into one with the weight of the new fact (c) computed by marginal inference.*

Algorithm 1 Coalescing bitemporal probabilistic KG.

procedure COALESCE(K)
  Input: uncoalecsed bitemporal KG $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$
  Output: coalesced KG K_c
  $\mathsf{N} \leftarrow$ all the concepts and relations in K
  for each concept and relation $m \in \mathsf{N}$ **do**
    $r_m \leftarrow$ create a rule using (r_{14})
    add r_m to $\mathcal{M}$
  end for
  $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{M}$
  repeat
    ground the rules in $\mathcal{F}$ using the assertions in $\mathcal{A}$ (see Algorithm 2)
    $\mathcal{A}' \leftarrow$ gather coalesced/inferred assertions
  until closure
  compute the marginal probabilities of the coalesced facts $\mathcal{A}'$ (Algorithm 2)
  use sort-merge to coalesce assertions in $\mathcal{A}$ and $\mathcal{A}'$ (Bohlen et al. [2])
  delete the uncoalesced ones in $\mathcal{A}$ and $\mathcal{A}$'
  return $K_c = (\mathcal{S}, \mathcal{A} \cup \mathcal{A}', \mathcal{F} \setminus \mathcal{M})$
end procedure

5 Inference

In this section, we present two important reasoning tasks in bitemporal probabilistic KG, namely, marginal and maximum a-posteriori (MAP) inference. We denote the set of constants in a bitemporal probabilistic KG $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ by $\mathcal{C}$, $\mathcal{C} \subseteq \mathsf{IL} \cup \mathsf{T}$ as the union of sets of IRIs, literals and time points. The Herbrand base $\mathrm{HB}(\mathcal{F})$ of $\mathcal{F}$ can be constructed by instantiating all the variables in $\mathcal{F}$ using the constants in $\mathcal{C}$ (aka. *grounding* or *expansion*). The function θ, given a finite set $\mathcal{C}$, maps each fact in some bitemporal KG into a subset of the Herbrand base $\mathrm{HB}(\mathcal{F})$ of $\mathcal{F}$ with respect to $\mathcal{C}$. Each subset of the Herbrand base is a Herbrand interpretation specifying which ground atoms are true. A Herbrand interpretation H is a Herbrand model of $\mathcal{F}$, denoted by $\models_H \mathcal{F}$, iff it satisfies all groundings of the formulas in $\mathcal{F}$.

▶ **Definition 5.** *Given $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ over a set of constants $\mathcal{C}$ and $\mathrm{HB}(\mathcal{F})$ the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$, $\theta : \mathcal{S} \cup \mathcal{A} \rightarrow \mathrm{HB}(\mathcal{F})$ maps $\mathcal{S} \cup \mathcal{A}$ into subsets of $\mathrm{HB}(\mathcal{F})$ as follows:*

$$\theta(\mathcal{S} \cup \mathcal{A}) = \bigcup\nolimits_{y \in \mathcal{S} \cup \mathcal{A}} \varphi(y),$$

$\varphi()$ maps axioms and assertions into FOL predicates using Definition 5. Besides, we extend θ as follows $\theta(K) = \theta(\mathcal{S} \cup \mathcal{A}) \cup \mathcal{F}$ where θ is a bijective function, it induces a one-to-one correspondence between the Herbrand models of $\mathcal{F}$ and *expanded* KG. As already discussed, φ maps inclusion axioms and instance assertions into first-order predicates. Grounding $\mathcal{F}$ with the constants of a bitemporal probabilistic KG may generate a set of new facts; this results in an *expanded* KG.

▶ **Lemma 2.** *Let $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ be a bitemporal probabilistic KG; let $\mathcal{C}$ be a set of constants; and let $HB(\mathcal{F})$ be the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$. We have two cases:*

- *for any $K' \subseteq K$, $K \models_{v,t} K' \Rightarrow \theta(K') \models_H \mathcal{F}$ and*
- *for any $H \subseteq HB$, $H \models_H \mathcal{F} \Rightarrow \theta^{-1}(H) \models_{v,t} K''$ and $K'' \subseteq K$.*

5.1 Marginal Inference

Marginal inference is the task of computing the marginal distributions of a set of random variables (queries). In our setting, given a query q and a bitemporal probabilistic KG K, marginal inference involves in computing the probability of the answers of q over K. In this paper, we are interested in Boolean temporal conjunctive queries.

▶ **Definition 6.** *A Boolean temporal conjunctive query q is a formula of the form*

$$q \leftarrow r_1(\mathbf{x}_1, \mathbf{v}_1, \mathbf{t}_1), \ldots, r_i(\mathbf{x}_i, \mathbf{v}_i, \mathbf{t}_i), \ldots, r_n(\mathbf{x}_n, \mathbf{v}_n, \mathbf{t}_n)$$

where $\mathbf{x}_i$ are atemporal variables or constants that are in r_i,

- *$\mathbf{v}_i$ are valid time variables or time points that are in r_i,*
- *$\mathbf{t}_i$ are transaction time variables or time points that are in r_i, and*
- *r_i denotes a temporal relation $\textsc{ca}(x_i, y_i, v_b, v_e, t_b, t_e)$ or $\textsc{ra}(x_i, y_i, v_b, v_e, t_b, t_e)$ with $x_i, y_i \in \mathbf{x}_i$, $v_b, v_e \in \mathbf{v}_i$ and $t_b, t_e \in \mathbf{t}_i$.*

In probabilistic databases and statistical relational learning, often the probabilities of queries are computed by grounding, i.e., by replacing all the variables in the queries using constants in the database. The grounding is used to generate a propositional sentence (lineage of a query) for exact inference or a graphical model for approximate computation. Similarly, Boolean temporal conjunctive queries can be grounded by evaluating queries using the *techniques from temporal databases* [16] and instantiating or replacing all the variables in the queries using their answers. This results in a set of ground queries, the probability of which can be computed using MLN systems or any other approximate inference engine. For Boolean temporal conjunctive queries, it is necessary to take care of the overlap of time intervals during grounding. One solution to this problem is to rewrite queries to take into account overlaps. In other terms, Boolean temporal conjunctive queries require checking interval intersection to determine the overlap of intervals in the query predicates. In order to do this, we rewrite queries as discussed below.

The marginal probability of a query q is obtained by a two-step process: (i) firstly, rewrite the query q, and (ii) secondly, perform marginal computation. To rewrite q we add a function called $\textsc{ov}$ (overlaps) which tests if both valid time and transaction times of relations are overlapping. Formally, the rewriting of a Boolean temporal query q:

$$q \leftarrow r_1(\mathbf{x}_1, \mathbf{v}_1, \mathbf{t}_1), \ldots, r_i(\mathbf{x}_i, \mathbf{v}_i, \mathbf{t}_i), \ldots, r_n(\mathbf{x}_n, \mathbf{v}_n, \mathbf{t}_n),$$

is the following:

$$q_r \leftarrow r_1(\mathbf{x}_1, \mathbf{v}_1, \mathbf{t}_1), \ldots, r_i(\mathbf{x}_i, \mathbf{v}_i, \mathbf{t}_i), \ldots, r_n(\mathbf{x}_n, \mathbf{v}_n, \mathbf{t}_n), \textsc{ov}(\mathbf{v}_1, \ldots, \mathbf{v}_n), \textsc{ov}(\mathbf{t}_1, \ldots, \mathbf{t}_n),$$

where $\textsc{ov}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ (resp. $\textsc{ov}(\mathbf{t}_1, \ldots, \mathbf{t}_n)$) is a Boolean function that tests if the valid (resp. transaction) time intervals $\mathbf{v}_1, \ldots, \mathbf{v}_n$ (resp. $\mathbf{t}_1, \ldots, \mathbf{t}_n$) are overlapping.

▶ **Definition 7.** *Given a Boolean temporal conjunctive query q and a bitemporal probabilistic KG $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$, the probability of q over K is computed using the following:*

$$P(q|\mathcal{S}, \mathcal{A}, \mathcal{F}) = Z^{-1}\exp\Big(\sum_{(f_j, w_j) \in \mathcal{F}: q_r \models_{v,t} f_j} w_j n(f_j, q_r, \mathcal{S}, \mathcal{A})\Big),$$

where q_r is the rewriting of q and $\models_{v,t}$ is bitemporal entailment. Note that computing Z takes exponential time in the worst case.

Algorithm 2 Approximate query probability.

1: **procedure** APPROXPROBABILITY(K,q,N)
2: **Input:** $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$, query q and hop N
3: **Output:** $P_a(q)$
4: $q_r \leftarrow$ rewrite(q)
5: Schema and assertion tables $T_\mathcal{S}$, $T_\mathcal{A} \leftarrow$ load($\mathcal{S}, \mathcal{A}$),
6: Rules table $T_\mathcal{F} \leftarrow$ load($\mathcal{F}$),
7: **for** $f_i \in T_\mathcal{F}$ **do** $T_K \leftarrow T_\mathcal{S} \bowtie T_\mathcal{A} \bowtie f_i$ **end for**
8: Answers table $T_q \leftarrow$ evaluate query(q_r, T_K)
9: Factor graph, $G_F \leftarrow \emptyset$
10: **for** each inference rule $f_i \in T_\mathcal{F}$ **do**
11: $G_F \leftarrow G_F \cup_B (T_q \bowtie f_i)$
12: **end for**
13: $G_F^N \leftarrow$ extract n-hop subgraph(G_F, T_q, N)
14: $P_a(q) \leftarrow$ compute(G_F^N,T_q)
15: **end procedure**

The complexity of computing the probability of a query is known to be #P-hard in general. The above definition shows that temporal scoping of instance assertions does not increase the complexity (since the OV function can be computed during ground which is before probability computation). This is consistent with the results in temporal databases, as already shown a temporal query has the same properties as that of a first-order query langauge [42].

5.1.1 Approximate Marginal Inference

Since exact marginal probability computation is an expensive operation (see the experimental results in Figure 3), often approximate sampling techniques are used to speed up inference. In general, a large portion of a KG is not relevant, for computing the probability of a given query, because in a Markov network a node/variable is independent of all the other nodes/variables given its neighboring nodes/variables (known as the Markov blanket). More formally, given a query variable x_i, $P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|\text{MB}(x_i))$ where $\text{MB}(x_i)$ is the Markov blanket of x_i. Thus for a given a query, a part of a KG up to a certain depth N, containing the query node, can be extracted to estimate the probability of that query. This approach is described in Algorithm 2. The algorithm takes as input a KG K, a query q and hop distance N. The atemporal schema and temporal assertions are loaded into relational database tables[3] $T_\mathcal{S}$ and $T_\mathcal{A}$ respectively. Likewise, the inference rules are loaded into the table $T_\mathcal{F}$. After loading the KG, we perform KG expansion by applying the inference rules until closure (line 7). Then after, we evaluate the rewriting q_r of the query q on the expanded KG T_K and keep its answers in a relational table T_q (line 8). In lines 9–12, we create a factor graph G_F by iteratively joining the answers table T_q with that of inference rules; $\cup_B$ denotes a bag unions operation. An N-hop subgraph − all nodes that are N distance away from the query node − G_F^N is extracted from G_F with a depth of N hops from the query nodes T_q (line 13). Finally, in line 14, we compute the approximate probability $P_a(q)$ of q using either exact or approximate (by using a Gibbs sampler) inference. Lifted inference is a focus of recent research which leverages the structure (e.g. using symmetries) of MLN

[3] Note that we exclude the discussion of the schema of the tables and SQL join queries for brevity.

knowledge bases for efficient inference. It is worthwhile comparing lifted inference with that of approximate marginal inference as well as investigating the possibility for a combined approach. We leave out this task as a future work.

5.2 MAP Inference

In bitemporal probabilistic KG, MAP inference is the problem of computing the most probable, consistent, and non-probabilistic bitemporal KG (also known as MAP state). More formally, given a KG $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ and a mapping function θ, we denote the MAP problem by $map(\theta(K))$. In order to compute $map(\theta(K))$, we need to translate K with the function θ into an equivalent MLN formalization. Then, the inference rules $\mathcal{F}$ are added to this translation. The MAP state is obtained by using $\theta(K)$ and the grounding of $\mathcal{F}$ as input. Applying the inverse translation function θ^{-1} to the MAP state, yields the most probable bitemporal KG.

▶ **Lemma 3.** *Let $K = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ be a bitemporal probabilistic KG, $\mathcal{C}$ a set of constants, and the Herbrand base $HB(\mathcal{F})$ of $\mathcal{F}$ with respect to $\mathcal{C}$. The MAP state of K is obtained as:*

$$\theta^{-1}(H) = \underset{H \subseteq HB : H \models_H \mathcal{F}}{\arg\max} \left(\sum_{(f,w_j) \in \mathcal{F} : H \models_{v,t} f_j} w_j n(f_i, H) \right)$$

Using the above lemma, it can be proved that computing the most probable and consistent bitemporal KG is NP-hard in general. This is shown by exploiting the correspondence between a bitemporal probabilistic KG and an MLN program, and using the bijective function θ. Membership is shown by mapping MAP inference in a bitemporal probabilistic KG into MAP inference in MLN. Hardness is shown by translating MLN programs into KG. From Lemma 3 and the results in [5] it follows that the problem of computing a MAP state is NP-hard.

6 Empirical Evaluation

We conducted two different kinds of experiments: (i) marginal and (ii) MAP inference. For both experiments, we carried out performance tests in terms of running times and the accuracy of marginal distributions. We run the experiments on a Debian 8 virtual machine with 2.6GHz 3-core Intel Haswell processor, 24 GB of main memory and 1TB disk space.

Tools. We used the following probabilistic reasoners: ProbLog [25], Tuffy [30], and TuffyLite [24]. *ProbLog* is a probabilistic extension of Prolog. Unlike MLN, ProbLog assigns probabilities to clauses (first order Horn formulas) and poses the restriction that these probabilities are mutually independent. It defines a probability distribution over logic programs [25]. Its semantics is defined by the success probability of a query in a randomly sampled program. On the other hand, *TuffyLite and Tuffy* are probabilistic reasoners for MLN programs. TuffyLite is an improved version of Tuffy. Hence, we use it for comparison with ProbLog. Note that we intentionally leave out DeepDive [38] from out experiments, because DeepDive uses the same technique as Tuffy and TuffyLite.

Data. For our experiments, we use the Wikidata KG. In particular, we consider a part of the KG that contains structured temporal information. Hence, we extracted temporal facts for various fluents (time-varying relations) including: *member of sports team*, *educated at*, *occupation*, *spouse*, and so on. Overall, we extracted over 3.7 million temporal facts. All of these temporal facts are only interval timestampped (valid time). As Wikidata is a valid time KG, we extended it by adding transaction time intervals and probabilities to each fact in order to have a bitemporal probabilistic KG. We randomly generated transaction times and probabilities (> 0.5). Besides, as Wikidata is not complete, it does not contain valid times for all the facts. If the end time point of a fact is missing, we replace it with now.

Temporal rules. We designed 36 different bitemporal probabilistic inference rules (ProbLog definite clauses) based on the fluents of Wikidata. For ProbLog, the probabilities p_i of the rules are generated randomly and are set between 0.5 and 0.99. However, for the MLN solver, TuffyLite, we use the log odds of the probabilities as weights, i.e., $w_i = \ln \frac{p_i}{1-p_i}$.

6.1 Marginal Inference

In this experiment, we test the scalability of marginal inference. The results of the experiment are reported in Figure 3(a). As shown, ProbLog performs very well, it outperforms TuffyLite. The run time of marginal inference is almost constant when the number of facts is less than 20,000. This is because ProbLog uses *knowledge compilation* to speed up inference. The reported runtimes are averaged over 5 runs. For TuffyLite, we test the quality of approximate marginal probabilities, we plot the prior and marginal probabilities of 8 randomly selected queries in Figure 3(b). As it can be seen, the approximate marginals are very close to the prior probabilities with the highest absolute error close to 2% for query q_6.

6.2 MAP Inference

In this experiment, we report the running times of ProbLog on different data sizes. We exclude Tuffy and TuffyLite from this experiment due to scalability issues (after running for more than 24hs both systems did not terminate). ProbLog performs most probable explanation (MPE) inference which is slightly different from MAP inference. The runtimes, averaged over 5 runs, are reported in Figure 3(c). As it can be seen, the runtime of ProbLog does not increase linearly with respect to the size of the input data. This is due to each added incorrect bitemporal fact might be involved in a conflict resulting in a non trivial optimization problem. In order to test the scalability of MPE inference, we increased the data size upto 500,000. As expected, the running times were exceedingly high, i.e., upto several hours. When the datasize is 500,000, we interrupted the evaluation of MPE inference after a runtime of 3.63 hours. In both MAP and marginal inference, scalability is a big problem. We will tackle this problem in the future.

7 Related work

In relational databases, bitemporal databases have been throughly invesitgated [27]. Besides, temporal databases have been extensively studied (see surveys [31, 39]). However, relatively fewer works exist on temporal probabilistic databases [11, 8]. In [11], a relational database is used to model and query temporal data, integrity constraints and deduction rules can also be specified. However, these rules must be deterministic (unweighted) unlike what we do here. On the other hand, contrary to this study where we use a bitemporal model, uncertain spatio-temporal databases focus on stochastically modelling trajectories through space and time (see [14] for instance).

Query evaluation in probabilistic databases is an active area of research [19, 7, 40, 12]. With respect to temporal query evaluation over a valid time probabilistic KG, to the best of our knowledge, there are two important studies [6] and [11]. While the former focuses on

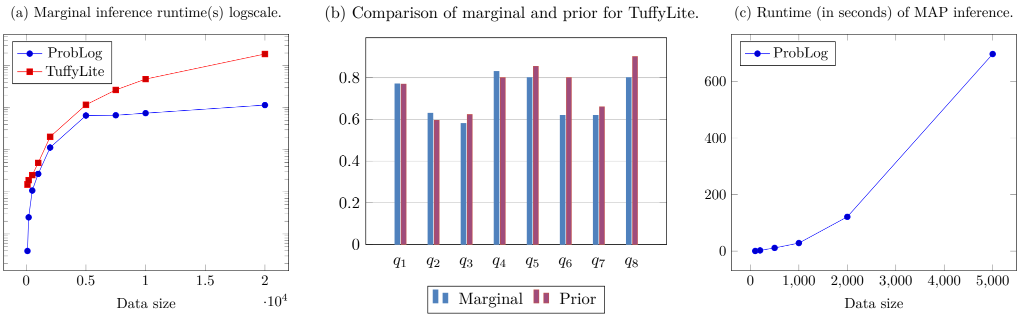


Figure 3 Performance of TuffyLite and ProbLog over fixed query and varying data size.

MAP inference, we study here both marginal and MAP inference in a bitemporal setting, besides, we deal with the problem of temporal coalescing and use a rich schema based on OWL RL. The later deals with marginal inference, the difference with this work are the following: (i) we consider weighted temporal OWL RL inference rules, (ii) we propose coalescing for bitemporal KG, and (iii) we introduce rewriting for coalescing of query answers. In another study [13], the authors proposed an approach for resolving temporal conflicts in RDF knowledge graphs. The idea is to use first-order logic Horn formulas with temporal predicates to express temporal and non-temporal constraints. However, these approaches are limited to a small set of temporal patterns and only allow for uncertainty in facts. Moreover, extending KG using open domain information extraction, will often also lead to uncertainty about the correctness of schema information; a large variety of inference rules and constraints, some of which will be domain specific, can also be the subject of uncertainty.

Multi-temporal RDF database models are first introduced by Grandi [18]. These models allow to capture several aspects of time in data such as validity, efficacy, transaction and so on in a non-probabilistic setting. By contrast, we consider validity and transaction times with a rich probabilistic schema.

8 Conclusion and Outlook

We have proposed a bitemporal model to assert the times in which facts are considered valid and the times when facts are added to a KG. Besides, we studied bitemporal probabilistic KG and proved that standard reasoning tasks such as marginal and MAP inference do not introduce any additional complexity. We have also introduced an efficient algorithm, for marginal inference, based on N-hop graph neighborhoods of query nodes. Furthermore, we have addressed coalescing in a probabilistic setting both in data and during query answering. Our experimental results show that scalability is challenging and approximate techniques with acceptable error margins can be adopted. As a future work, we plan to do further experiments, testing scalability as well as the accuracy of state-of-the-art reasoners and implementation of our approximate marginal inference algorithm.

References

1 Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

2 Michael H. Böhlen, Richard T. Snodgrass, and Michael D. Soo. Coalescing in Temporal Databases. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India*, pages 180–191, 1996.

3 Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, volume 5, page 3, 2010.

4 Melisachew Wudage Chekol, Valeria Fionda, and Giuseppe Pirrò. Time Travel Queries in RDF Archives. In *Joint proceedings of the 3rd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2017) and the 4th Workshop on Linked Data Quality (LDQ 2017) co-located with 14th European Semantic Web Conference (ESWC 2017), Portorož, Slovenia, May 28th-29th, 2017.*, pages 28–42, 2017.

5 Melisachew Wudage Chekol, Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. Markov Logic Networks with Numerical Constraints. In *ECAI 2016*, pages 1017–1025, 2016.

6 Melisachew Wudage Chekol, Giuseppe Pirrò, Joerg Schoenfisch, and Heiner Stuckenschmidt. Marrying Uncertainty and Time in Knowledge Graphs. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 88–94, 2017.

7 Yang Chen, Daisy Zhe Wang, and Sean Goldberg. ScaLeKB: scalable learning and inference over large knowledge bases. *The VLDB Journal*, 25(6):893–918, 2016.

8 Alex Dekhtyar, Robert Ross, and VS Subrahmanian. Probabilistic temporal databases, I: algebra. *ACM Transactions on Database Systems (TODS)*, 26(1):41–95, 2001.

9 Anton Dignös, Michael H Böhlen, and Johann Gamper. Temporal alignment. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 433–444. ACM, 2012.

10 Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *SIGKDD*, pages 601–610, 2014.

11 Maximilian Dylla, Iris Miliaraki, and Martin Theobald. A temporal-probabilistic database model for information extraction. *Proc. of the VLDB Endowment*, 6(14):1810–1821, 2013.

12 Maximilian Dylla, Iris Miliaraki, and Michael Theobald. Top-k query processing in probabilistic databases with non-materialized views. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 122–133. IEEE, 2013.

13 Maximilian Dylla, Mauro Sozio, and Martin Theobald. Resolving Temporal Conflicts in Inconsistent RDF Knowledge Bases. In *BTW*, pages 474–493, 2011.

14 Tobias Emrich, Hans-Peter Kriegel, Nikos Mamoulis, Matthias Renz, and Andreas Zufle. Querying uncertain spatio-temporal data. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 354–365. IEEE, 2012.

15 Opher Etzion. *Temporal databases: research and practice*, volume 1399. Springer Science & Business Media, 1998.

16 Dengfeng Gao, S Jensen, T Snodgrass, and D Soo. Join operations in temporal databases. *The VLDB Journal – The International Journal on Very Large Data Bases*, 14(1):2–29, 2005.

17 Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. Minie: minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, 2017.

18 Fabio Grandi. Multi-temporal RDF ontology versioning. In *Proceedings of the 3rd International Workshop on Ontology Dynamics (IWOD-09)*, 2009.

19 Eric Gribkoff and Dan Suciu. SlimShot: In-Database Probabilistic Inference for Knowledge Bases. *PVLDB*, 9(7):552–563, 2016.

20 Claudio Gutierrez, Carlos A Hurtado, and Alejandro Vaisman. Introducing time into RDF. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):207–218, 2007.

21 Olaf Hartig and Bryan Thompson. Foundations of an alternative approach to reification in RDF. *arXiv preprint*, 2014. `arXiv:1406.3399`.

22 Patrick Hayes. RDF Semantics. W3C Recommendation, 2004 .

23 Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, pages 229–232. ACM, 2011.

24 Tushar Khot, Niranjan Balasubramanian, Eric Gribkoff, Ashish Sabharwal, Peter Clark, and Oren Etzioni. Markov logic networks for natural language question answering. *arXiv preprint*, 2015. `arXiv:1507.03045`.

25 Angelika Kimmig, Bart Demoen, Luc De Raedt, Vitor Santos Costa, and Ricardo Rocha. On the implementation of the probabilistic logic programming language ProbLog. *Theory and Practice of Logic Programming*, 11(2-3):235–262, 2011.

26 Markus Krötzsch. OWL 2 Profiles: An introduction to lightweight ontology languages. In Thomas Eiter and Thomas Krennwallner, editors, *Proceedings of the 8th Reasoning Web Summer School, Vienna, Austria, September 3–8 2012*, volume 7487 of *LNCS*, pages 112–183. Springer, 2012.

27 Anil Kumar, Vassilis J Tsotras, and Christos Faloutsos. Designing access methods for bitemporal databases. *IEEE Transactions on Knowledge and Data Engineering*, 10(1):1–20, 1998.

28 Xiao Ling and Daniel S Weld. Temporal Information Extraction. In *AAAI*, volume 10, pages 1385–1390, 2010.

29 Boris Motik. Representing and querying validity time in RDF and OWL: A logic-based approach. *J. Web Semantics*, 12:3–21, 2012.

30 Feng Niu, Christopher Ré, AnHai Doan, and Jude Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an RDBMS. *Proc. of the VLDB Endowment*, 4(6):373–384, 2011.

31 Gultekin Ozsoyoglu and Richard T Snodgrass. Temporal and real-time databases: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):513–532, 1995.

32 Katerina Papaioannou and Michael Böhlen. TemProRA: Top-k temporal-probabilistic results analysis. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*, pages 1382–1385. IEEE, 2016.

33 Luc De Raedt, Anton Dries, Ingo Thon, Guy Van den Broeck, and Mathias Verbeke. Inducing Probabilistic Relational Rules from Probabilistic Examples. In *IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1835–1843, 2015.

34 Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

35 Anisa Rula, Matteo Palmonari, Andreas Harth, Steffen Stadtmüller, and Andrea Maurino. On the diversity and availability of temporal information in linked open data. *The Semantic Web–ISWC 2012*, pages 492–507, 2012.

36 Anisa Rula, Matteo Palmonari, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Jens Lehmann, and Lorenz Bühmann. Hybrid acquisition of temporal scopes for RDF data. In *European Semantic Web Conference*, pages 488–503. Springer, 2014.

37 Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *EMNLP*, pages 1088–1098, 2010.

38 Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. Incremental knowledge base construction using deepdive. *Proceedings of the VLDB Endowment*, 8(11):1310–1321, 2015.

39 Richard T Snodgrass. Temporal databases. In *Theories and methods of spatio-temporal reasoning in geographic space*, pages 22–64. Springer, 1992.

40 Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

41 Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. Coupled temporal scoping of relational facts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 73–82. ACM, 2012.

42 David Toman. Point vs. interval-based query languages for temporal databases. In *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 58–67. ACM, 1996 .

43 Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

44 Esteban Zimányi. Temporal aggregates and temporal universal quantification in standard SQL. *ACM SIGMOD Record*, 35(2):16–21, 2006.